Does Baum-Welch Re-estimation Help Taggers?

David Elworthy
Sharp Laboratories of Europe Ltd.
Edmund Halley Road
Oxford Science Park
Oxford OX4 4GA
United Kingdom
dahe@sharp.co.uk

Abstract

In part of speech tagging by Hidden Markov Model, a statistical model is used to assign grammatical categories to words in a text. Early work in the field relied on a corpus which had been tagged by a human annotator to train the model. More recently, Cutting *et al.* (1992) suggest that training can be achieved with a minimal lexicon and a limited amount of *a priori* information about probabilities, by using Baum-Welch re-estimation to automatically refine the model. In this paper, I report two experiments designed to determine how much manual training information is needed. The first experiment suggests that initial biasing of either lexical or transition probabilities is essential to achieve a good accuracy. The second experiment reveals that there are three distinct patterns of Baum-Welch re-estimation. In two of the patterns, the re-estimation ultimately reduces the accuracy of the tagging rather than improving it. The pattern which is applicable can be predicted from the quality of the initial model and the similarity between the tagged training corpus (if any) and the corpus to be tagged. Heuristics for deciding how to use re-estimation in an effective manner are given. The conclusions are broadly in agreement with those of Merialdo (1994), but give greater detail about the contributions of different parts of the model.

Citation details: appears in Proceedings of the 4th ACL Conference on Applied Natural Language Processing, Stuttgart, October 13-15th 1994, pp. 53–58. Some typos corrected from the published version.

1 Background

Part-of-speech tagging is the process of assigning grammatical categories to individual words in a corpus. One widely used approach makes use of a statistical technique called a Hidden Markov Model (HMM). The model is defined by two collections of parameters: the *transition probabilities*, which express the probability that a tag follows the preceding one (or two for a second order model); and the *lexical probabilities*, giving the probability that a word has a given tag without regard to words on either side of it. To tag a text, the tags with non-zero probability are hypothesised for each word, and the most probable sequence of tags given the sequence of words is determined from the probabilities. Two algorithms are commonly used, known as the Forward-Backward (FB) and Viterbi algorithms. FB assigns a probability to every tag on every word, while Viterbi prunes tags which cannot be chosen because their probability is lower than the ones of competing hypotheses, with a corresponding gain in computational efficiency. For an introduction to the algorithms, see Cutting *et al.* (1992), or the lucid description by Sharman (1990).

There are two principal sources for the parameters of the model. If a tagged corpus prepared by a human annotator is available, the transition and lexical probabilities can be estimated from the frequencies of pairs of tags and of tags associated with words. Alternatively, a procedure called Baum-Welch (BW) re-estimation may be used, in which an untagged corpus is passed through the FB algorithm with some initial model, and the resulting probabilities used to determine new values for the lexical and transition probabilities. By iterating the algorithm with the same corpus, the parameters of the model can be made to converge on values which are locally optimal for the given text. The degree of convergence can be measured using a perplexity measure, the sum of $p\log_2 p$ for hypothesis probabilities

p, which gives an estimate of the degree of disorder in the model. The algorithm is again described by Cutting *et al.* and by Sharman, and a mathematical justification for it can be found in Huang *et al.* (1990).

The first major use of HMMs for part of speech tagging was in CLAWS (Garside *et al.*, 1987) in the 1970s. With the availability of large corpora and fast computers, there has been a recent resurgence of interest, and a number of variations on and alternatives to the FB, Viterbi and BW algorithms have been tried; see the work of, for example, Church (Church, 1988), Brill (Brill and Marcus, 1992; Brill, 1992), DeRose (DeRose, 1988) and Kupiec (Kupiec, 1992). One of the most effective taggers based on a pure HMM is that developed at Xerox (Cutting *et al.*, 1992). An important aspect of this tagger is that it will give good accuracy with a minimal amount of manually tagged training data. 96% accuracy correct assignment of tags to word token, compared with a human annotator, is quoted, over a 500000 word corpus.

The Xerox tagger attempts to avoid the need for a hand-tagged training corpus as far as possible. Instead, an approximate model is constructed by hand, which is then improved by BW re-estimation on an untagged training corpus. In the above example, 8 iterations were sufficient. The initial model set up so that some transitions and some tags in the lexicon are favoured, and hence having a higher initial probability. Convergence of the model is improved by keeping the number of parameters in the model down. To assist in this, low frequency items in the lexicon are grouped together into equivalence classes, such that all words in a given equivalence class have the same tags and lexical probabilities, and whenever one of the words is looked up, then the data common to all of them is used. Re-estimation on any of the words in a class therefore counts towards re-estimation for all of them[1].

The results of the Xerox experiment appear very encouraging. Preparing tagged corpora by hand is labour-intensive and potentially error-prone, and although a semi-automatic approach can be used (Marcus *et al.*, 1993), it is a good thing to reduce the human involvement as much as possible. However, some careful examination of the experiment is needed. In the first place, Cutting *et al.* do not compare the success rate in their work with that achieved from a hand-tagged training text with no re-estimation. Secondly, it is unclear how much the initial biasing contributes the success rate. If significant human intervention is needed to provide the biasing, then the advantages of automatic training become rather weaker, especially if such intervention is needed on each new text domain. The kind of biasing Cutting *et al.* describe reflects linguistic insights combined with an understanding of the predictions a tagger could reasonably be expected to make and the ones it could not.

The aim of this paper is to examine the role that training plays in the tagging process, by an experimental evaluation of how the accuracy of the tagger varies with the initial conditions. The results suggest that a completely unconstrained initial model does not produce good quality results, and that one accurately trained from a hand-tagged corpus will generally do better than using an approach based on re-estimation, even when the training comes from a different source. A second experiment shows that there are different patterns of re-estimation, and that these patterns vary more or less regularly with a broad characterisation of the initial conditions. The outcome of the two experiments together points to heuristics for making effective use of training and re-estimation, together with some directions for further research.

Work similar to that described here has been carried out by Merialdo (1994), with broadly similar conclusions. We will discuss this work below. The principal contribution of this work is to separate the effect of the lexical and transition parameters of the model, and to show how the results vary with different degree of similarity between the training and test data.

2 The tagger and corpora

The experiments were conducted using two taggers, one written in C at Cambridge University Computer Laboratory, and the other in C++ at Sharp Laboratories. Both taggers implement the FB, Viterbi and BW algorithms. For training from a hand-tagged corpus, the model is estimated by counting the number of transitions from each tag i to each tag j, the total occurrence of each tag i, and the total occurrence of word w with tag i. Writing these as $f(i,j)$, $f(i)$ and $f(i,w)$ respectively, the transition probability from tag i to tag j is estimated as $f(i,j)/f(i)$ and the lexical probability as $f(i,w)/f(i)$. Other estimation formulae have been used in the past. For example, CLAWS (Garside *et al.*, 1987) normalises the lexical probabilities by the total frequency of the word rather than of the tag. Consulting the Baum-Welch re-estimation formulae suggests that the approach described is more appropriate, and this is confirmed by slightly greater tagging accuracy. Any

[1]The technique was originally developed by Kupiec (Kupiec, 1989).

transitions not seen in the training corpus are given a small, non-zero probability.

The lexicon lists, for each word, all of tags seen in the training corpus with their probabilities. For words not found in the lexicon, all open-class tags are hypothesised, with equal probabilities. These words are added to the lexicon at the end of first iteration when re-estimation is being used, so that the probabilities of their hypotheses subsequently diverge from being uniform.

To measure the accuracy of the tagger, we compare the chosen tag with one provided by a human annotator. Various methods of quoting accuracy have been used in the literature, the most common being the proportion of words (tokens) receiving the correct tag. A better measure is the proportion of *ambiguous* words which are given the correct tag, where by ambiguous we mean that more than one tag was hypothesised. The former figure looks more impressive, but the latter gives a better measure of how well the tagger is doing, since it factors out the trivial assignment of tags to non-ambiguous words. For a corpus in which a fraction a of the words are ambiguous, and p is the accuracy on ambiguous words, the overall accuracy can be recovered from $1 - a + pa$. All of the accuracy figures quoted below are for ambiguous words only.

The training and test corpora were drawn from the LOB corpus and the Penn treebank. The hand tagging of these corpora is quite different. For example, the LOB tagset used 134 tags, while the Penn treebank tagset has 48. The general pattern of the results presented does not vary greatly with the corpus and tagset used.

3 The effect of the initial conditions

The first experiment concerned the effect of the initial conditions on the accuracy using Baum-Welch re-estimation. A model was trained from a hand-tagged corpus in the manner described above, and then degraded in various ways to simulate the effect of poorer training, as follows:

Lexicon

- **D0** Un-degraded lexical probabilities, calculated from $f(i, w)/f(i)$.
- **D1** Lexical probabilities are correctly ordered, so that the most frequent tag has the highest lexical probability and so on, but the absolute values are otherwise unreliable.
- **D2** Lexical probabilities are proportional to the overall tag frequencies, and are hence independent of the actual occurrence of the word in the training corpus.
- **D3** All lexical probabilities have the same value, so that the lexicon contains no information other than the possible tags for each word.

Transitions

- **T0** Un-degraded transition probabilities, calculated from $f(i, j)/f(i)$.
- **T1** All transition probabilities have the same value.

We could expect to achieve D1 from, say, a printed dictionary listing parts of speech in order of frequency. Perfect training is represented by case D0+T0. The Xerox experiments (Cutting *et al.*, 1992) correspond to something between D1 and D2, and between T0 and T1, in that there is some initial biasing of the probabilities.

For the test, four corpora were constructed from the LOB corpus: LOB-B from part B, LOB-L from part L, LOB-B-G from parts B to G inclusive and LOB-B-J from parts B to J inclusive. Corpus LOB-B-J was used to train the model, and LOB-B, LOB-L and LOB-B-G were passed through thirty iterations of the BW algorithm as untagged data. In each case, the best accuracy (on ambiguous words, as usual) from the FB algorithm was noted. As an additional test, we tried assigning the most probable tag from the D0 lexicon, completely ignoring tag-tag transitions. The results are summarised in table 1, for various corpora, where F denotes the "most frequent tag" test. As an example of how these figures relate to overall accuracies, LOB-B contains 32.35% ambiguous tokens with respect to the lexicon from LOB-B-J, and the overall accuracy in the D0+T0 case is hence 98.69%. The general pattern of the results is similar across the three test corpora, with the only difference of interest being that case D3+T0 does better for LOB-L than for the other two cases, and in particular does better than cases D0+T1 and D1+T1. A possible explanation is that in this case the test data does not overlap with the training data, and hence the good quality lexicons (D0 and D1) have less of an influence. It is also interesting that D3+T1 does better than D2+T1. The reasons for this are unclear, and the results are not always the same with other corpora, which suggests that they are not statistically significant.

Several follow-up experiments were used to confirm the results: using corpora from the Penn treebank, using equivalence classes to ensure that all lexical entries have a total relative frequency of at least 0.01, and using larger corpora. The specific accuracies were different in the various tests, but the

Table 1: Accuracy using Baum-Welch re-estimation with various initial conditions

Dict	Trans	LOB-B (%)	LOB-L (%)	LOB-B-G (%)
D0	T0	95.96	94.77	96.17
D1	T0	95.40	94.44	95.40
D2	T0	90.52	91.82	92.36
D3	T0	92.96	92.80	93.48
D0	T1	94.06	92.27	94.51
D1	T1	94.06	92.27	94.51
D2	T1	66.51	72.48	55.88
D3	T1	75.49	80.87	79.12
F	-	89.22	85.32	88.71

overall patterns remained much the same, suggesting that they are not an artifact of the tagset or of details of the text.

The observations we can make about these results are as follows. Firstly, two of the tests, D2+T1 and D3+T1, give very poor performance. Their accuracy is not even as good as that achieved by picking the most frequent tag (although this of course implies a lexicon of D0 or D1 quality). It follows that if Baum-Welch re-estimation is to be an effective technique, the initial data must have either biasing in the transitions (the T0 cases) or in the lexical probabilities (cases D0+T1 and D1+T1), but it is not necessary to have both (D2/D3+T0 and D0/D1+T1).

Secondly, training from a hand-tagged corpus (case D0+T0) always does best, even when the test data is from a different source to the training data, as it is for LOB-L. So perhaps it is worth investing effort in hand-tagging training corpora after all, rather than just building a lexicon and letting re-estimation sort out the probabilities. But how can we ensure that re-estimation will produce a good quality model? We look further at this issue in the next section.

4 Patterns of re-estimation

During the first experiment, it became apparent that Baum-Welch re-estimation sometimes decreases the accuracy as the iteration progresses. A second experiment was conducted to decide when it is appropriate to use Baum-Welch re-estimation at all. There seem to be three patterns of behaviour:

Classical A general trend of rising accuracy on each iteration, with any falls in accuracy being local. It indicates that the model is converging towards an optimum which is better than its starting point.

Initial maximum Highest accuracy on the first iteration, and falling thereafter. In this case the initial model is of better quality than BW can achieve. That is, while BW will converge on an optimum, the notion of optimality is with respect to the HMM rather than to the linguistic judgements about correct tagging.

Early maximum Rising accuracy for a small number of iterations (2–4), and then falling as in initial maximum.

An example of each of the three behaviours is shown in figure 1. The values of the accuracies and the test conditions are unimportant here; all we want to show is the general patterns. The second experiment had the aim of trying to discover which pattern applies under which circumstances, in order to help decide how to train the model. Clearly, if the expected pattern is initial maximum, we should not use BW at all, if early maximum, we should halt the process after a few iterations, and if classical, we should halt the process in a "standard" way, such as comparing the perplexity of successive models.

The tests were conducted in a similar manner to those of the first experiment, by building a lexicon and transitions from a hand tagged training corpus, and then applying them to a test corpus with varying degrees of degradation. Firstly, four different degrees of degradation were used: no degradation at all, D2 degradation of the lexicon, T1 degradation of the transitions, and the two together. Secondly, we selected test corpora with varying degrees of similarity to the training corpus: the same text, text from a similar domain, and text which is significantly different. Two tests were conducted with each combination of the degradation and similarity, using different corpora (from the Penn treebank) ranging in size from approximately 50000 words to 500000 words. The re-estimation was allowed to run for ten iterations.

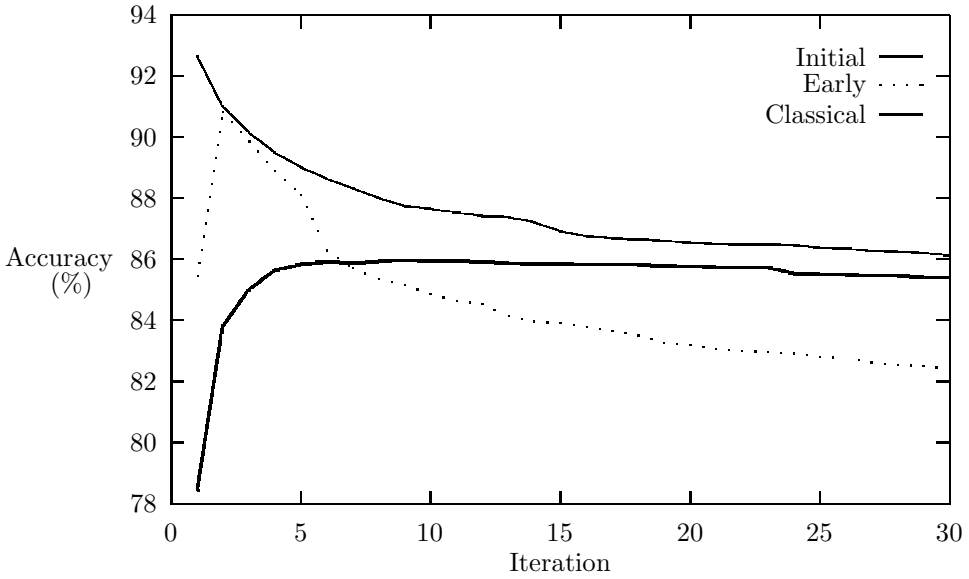

Figure 1: Example Baum-Welch behaviour

The results appear in table 2, showing the best accuracy achieved (on ambiguous words). the iteration at which it occurred, and the pattern of re-estimation (I = initial maximum, E = early maximum, C = classical). The patterns are summarised in table 3, each entry in the table showing the patterns for the two tests under the given conditions. Although there is some variations in the readings, for example in the "similar/D0+T0" case, we can draw some general conclusions about the patterns obtained from different sorts of data. When the lexicon is degraded (D2), the pattern is always classical. With a good lexicon but either degraded transitions or a test corpus differing from the training corpus, the pattern tends to be early maximum. When the test corpus is very similar to the model, then the pattern is initial maximum. Furthermore, examining the accuracies in table 2, in the cases of initial maximum and early maximum, the accuracy tends to be significantly higher than with classical behaviour. It seems likely that what is going on is that the model is converging to towards something of similar "quality" in each case, but when the pattern is classical, the convergence starts from a lower quality model and improves, and in the other cases, it starts from a higher quality one and deteriorates. In the case of early maximum, the few iterations where the accuracy is improving correspond to the creation of entries for unknown words and the fine tuning of ones for known ones, and these changes outweigh those produced by the re-estimation.

5 Discussion

From the observations in the previous section, we propose the following guidelines for how to train a HMM for use in tagging:

- If a hand-tagged training corpus is available, use it . If the test and training corpora are near-identical, do not use BW re-estimation; otherwise use for a small number of iterations.
- If no such training corpus is available, but a lexicon with at least relative frequency data is available, use BW re-estimation for a small number of iterations.
- If neither training corpus nor lexicon are available, use BW re-estimation with standard convergence tests such as perplexity. Without a lexicon, some initial biasing of the transitions is needed if good results are to be obtained.

Similar results are presented by Merialdo (1994), who describes experiments to compare the effect of training from a hand-tagged corpora and using the Baum-Welch algorithm with various initial conditions. As in the experiments above, BW re-estimation gave a decrease in accuracy when the starting point was derived from a significant amount of hand-tagged text. In addition, although Merialdo does not highlight the point, BW re-estimation

Table 2: Baum-Welch patterns (data)

Corpus relation	Degradation	Test 1			Test 2		
		Best (%)	at	pattern	Best (%)	at	pattern
Same	D0+T0	93.11	1	I	92.83	1	I
Similar	D0+T0	89.95	1	I	75.03	2	E
Different	D0+T0	84.59	2	E	86.00	2	E
Same	D0+T1	91.71	2	E	90.52	2	E
Similar	D0+T1	87.93	2	E	70.63	3	E
Different	D0+T1	80.87	3	E	82.68	3	E
Same	D2+T0	84.87	10	C	87.31	8	C
Similar	D2+T0	81.07	9	C	71.40	4	C*
Different	D2+T0	78.54	5	C*	80.81	9	C
Same	D2+T1	72.58	9	C	80.53	10	C
Similar	D2+T1	68.35	10	C	62.76	10	C
Different	D2+T1	65.64	10	C	68.95	10	C

* These tests gave an early peak, but the graphs of accuracy against number of iterations show the pattern to be classical rather than early maximum.

Table 3: Baum-Welch patterns (summary)

Degradation	D0+T0	D0+T1	D2+T0	D2+T1
Corpus relation				
Same	I, I	E, E	C, C	C, C
Similar	I, E	E, E	C, C	C, C
Different	E, E	E, E	C, C	C, C

starting from less than 5000 words of hand-tagged text shows early maximum behaviour. Merialdo's conclusion is that taggers should be trained using as much hand-tagged text as possible to begin with, and only then applying BW re-estimation with untagged text. The step forward taken in the work here is to show that there are three patterns of re-estimation behaviour, with differing guidelines for how to use BW effectively, and that to obtain a good starting point when a hand-tagged corpus is not available or is too small, either the lexicon or the transitions must be biased.

While these may be useful heuristics from a practical point of view, the next step forward is to look for an automatic way of predicting the accuracy of the tagging process given a corpus and a model. Some preliminary experiments with using measures such as perplexity and the average probability of hypotheses show that, while they do give an indication of convergence during re-estimation, neither shows a strong correlation with the accuracy. Perhaps what is needed is a "similarity measure" between two models M and M', such that if a corpus were tagged with model M, M' is the model obtained by training from the output corpus from the tagger as if it were a hand-tagged corpus. However, preliminary experiments using such measures as the Kullback-Liebler distance between the initial and new models have again showed that it does not give good predictions of accuracy. In the end it may turn out there is simply no way of making the prediction without a source of information extrinsic to both model and corpus.

Acknowledgements

The work described here was carried out at the Cambridge University Computer Laboratory as part of Esprit BR Project 7315 "The Acquisition of Lexical Knowledge" (Acquilex-II). The results were confirmed and extended at Sharp Laboratories of Europe. I thank Ted Briscoe for his guidance and advice, and the ANLP referees for their comments.

References

[Brill and Marcus1992] Eric Brill and Mitch Marcus (1992). Tagging an Unfamiliar Text With Minimal Human Supervision. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 10–16.

[Brill1992] Eric Brill (1992). A Simple Rule-Based Part of Speech Tagger. In *Third Conference on Applied Natural Language Processing. Proceedings of the Conference. Trento, Italy*, pages 152–155, Association for Computational Linguistics.

[Church1988] Kenneth Ward Church (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Second Conference on Applied Natural Language Processing. Proceedings of the Conference*, pages 136–143, Association for Computational Linguistics.

[Cutting *et al.*1992] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun (1992). A Practical Part-of-Speech Tagger. In *Third Conference on Applied Natural Language Processing. Proceedings of the Conference. Trento, Italy*, pages 133–140, Association for Computational Linguistics.

[DeRose1988] Steven J. DeRose (1988). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14(1):31–39.

[Garside *et al.*1987] Roger Garside, Geoffrey Leech, and Geoffrey Sampson (1987). *The Computational Analysis of English: A Corpus-based Approach*. Longman, London.

[Huang *et al.*1990] X. D. Huang, Y. Ariki, and M. A. Jack (1990). *Hidden Markov Models for Speech Recognition*. Edinburgh University Press.

[Kupiec1989] J. M. Kupiec (1989). Probabilistic Models of Short and Long Distance Word Dependencies in Running Text. In *Proceedings of the 1989 DARPA Speech and Natural Language Workshop*, pages 290–295.

[Kupiec1992] Julian Kupiec (1992). Robust Part-of-speech Tagging Using a Hidden Markov Model. *Computer Speech and Language*, 6.

[Marcus *et al.*1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[Merialdo1994] Bernard Merialdo (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171.

[Sharman1990] R. A. Sharman (1990). *Hidden Markov Model Methods for Word Tagging*. Technical Report UKSC 214, IBM UK Scientific Centre.